

Python: module `vcs.continents`

vcs.continents

[index](#)

Continents (Gcon) module

Modules

[vcs.Canvas](#)

[vcs.vcs](#)

[vcs.queries](#)

Classes

Gcon

class *Gcon*

Class: Gcon

Continents

Description of Gcon Class:

The Continents graphics method draws a predefined, generic set of outlines in a longitude by latitude space. (To draw continental external data set is required.)

This class is used to define an continents table entry used in VCS. It can be used to change some or all of the continents attributes in an existing continents table entry.

Other Useful Functions:

```
a=vcs.init()           # Constructor
a.show('continents')  # Show predefined boxfill continents
a.show('line')        # Show predefined line continents
a.setcolormap("AMIP") # Change the VCS color map
a.continents(c,'default') # Plot continents, where 'c' is a
                        # continents object
a.update()            # Updates the VCS Canvas with the
a.mode=1, or 0       # If 1, then automatic update. If
                        # 0, then use update function to
                        # update the VCS Canvas.
```

Example of Use:

```
a=vcs.init()
```

To Create a new instance of continents use:

```
con=a.createcontinents('new','quick')#copies content of 'quick' to
con=a.createcontinents('new') # copies content of 'default' to
```

To Modify an existing continents use:

```
con=a.getcontinents('AMIP_psl')

con.list() # Will list all the continents
con.projection='linear'
lon30={-180:'180W',-150:'150W',0:'Eq'}
con.xticlabels1=lon30
con.xticlabels2=lon30
con.xticlabels(lon30, lon30) # Will set them both
con.xmtics1=''
con.xmtics2=''
con.xmtics(lon30, lon30) # Will set them both
con.yticlabels1=lat10
con.yticlabels2=lat10
con.yticlabels(lat10, lat10) # Will set them both
con.ymtics1=''
con.ymtics2=''
con.ymtics(lat10, lat10) # Will set them both
con.datawc_y1=-90.0
con.datawc_y2=90.0
con.datawc_x1=-180.0
con.datawc_x2=180.0
con.datawc(-90, 90, -180, 180) # Will set them all
```

Specify the continents line style (or type):

```
con.line=0 # Same as con.line='solid'
con.line=1 # Same as con.line='dash'
con.line=2 # Same as con.line='dot'
con.line=3 # Same as con.line='dash-dot'
con.line=4 # Same as con.line='long-dash'
```

There are three possibilities for setting the line color indices

```
con.linecolor=22 # Same as con.linecolor=(22)
con.linecolor=(22) # Will set the continents to a specific color
con.linecolor=None # Turns off the line color index, default color
```

```
con.linewidth=1 # width range: 1 to 100, default color
```

Methods defined here:

```
__init__(self, parent, Gcon_name=None, Gcon_name_src='default', createGcon=0)
```

```
__setattr__(self, name, value)
```

```
datawc(self, dsp1=1e+20, dsp2=1e+20, dsp3=1e+20, dsp4=1e+20)
```

```
list(self)
```

```
script(self, script_filename=None, mode=None)
```

```
Function: script # Calls _vcs.s
```

Description of Function:

Saves out a continents graphics method in Python or VCS designated file.

Example of Use:

```
script(scriptfile_name, mode)
```

where: scriptfile_name is the output name of the
mode is either "w" for replace or "a" for

Note: If the the filename has a ".py" at the end
Python script. If the filename has a ".scr"
produce a VCS script. If neither extension
default a Python script will be produced.

```
a=vcs.init()  
con=a.createcontinents('temp')  
con.script('filename.py')           # Append to a Python fil  
con.script('filename.scr')         # Append to a VCS file "  
con.script('filename','w')
```

```
xmtics(self, xmt1=",", xmt2=")
```

```
xticlabels(self, xtl1=",", xtl2=")
```

```
ymtics(self, ymt1=",", ymt2=")
```

```
yticlabels(self, ytl1=",", ytl2=")
```

Functions

```
getGconmember(self, member)  
#####  
#  
# Function:      getGconmember  
#  
# Description of Function:  
#     Private function that retrieves the continents members from  
#     structure and passes it back to Python.  
#  
#  
# Example of Use:  
#     return_value =  
#     getGconmember(self, name)  
#         where: self is the class (e.g., Gcon)  
#                name is the name of the member that is being  
#  
#####
```

```
renameGcon(self, old_name, new_name)
```

```
#####
#
# Function:      renameGcon
#
# Description of Function:
#     Private function that renames the name of an existing cont
#     graphics method.
#
#
# Example of Use:
#     renameGcon(old_name, new_name)
#         where: old_name is the current name of continents g
#         new_name is the new name for the continents g
#
#####
```

setGconmember(self, member, value)

```
#####
#
# Function:      setGconmember
#
# Description of Function:
#     Private function to update the VCS canvas plot. If the can
#     set to 0, then this function does nothing.
#
#
# Example of Use:
#     setGconmember(self, name, value)
#         where: self is the class (e.g., Gcon)
#         name is the name of the member that is being
#         value is the new value of the member (or att
#
#####
```

Data

StringTypes = (<type 'str'>, <type 'unicode'>)